Tool support for refactoring functional programs

Full text Pdf (156 KB)

Source **Haskell Workshop** archive
Proceedings of the 2003 ACM SIGPLAN workshop on Haskell table of contents
Uppsala, Sweden
Pages: 27 - 38
Year of Publication: 2003
ISBN:1-58113-758-3

Authors Huiqing Li University of Kent
Claus Reinke University of Kent
Simon Thompson University of Kent

Sponsors SIGPLAN: ACM Special Interest Group on Programming Languages
ACM: Association for Computing Machinery

Publisher ACM Press New York, NY, USA

Additional Information: abstract references cited by index terms collaborative colleagues

Tools and Actions: Find similar Articles Review this Article
Save this Article to a Binder Display Formats: BibTex EndNote ACM Ref

DOI Bookmark: Use this link to bookmark this Article: http://doi.acm.org/10.1145/871895.871899
What is a DOI?

↑ ABSTRACT

Refactorings are source-to-source program transformations which change program structure and organisation, but not program functionality. Documented in catalogues and supported by tools, refactoring provides the means to adapt and improve the design of existing code, and has thus enabled the trend towards modern agile software development processes. Refactoring has taken a prominent place in software development and maintenance, but most of this recent success has taken place in the OO and XP communities.In our project, we explore the prospects for *'Refactoring Functional Programs'*, taking Haskell as a concrete case-study. This paper discusses the variety of pragmatic and implementation issues raised by our work on the *Haskell Refactorer*. We briefly introduce the ideas behind refactoring, and a set of elementary functional refactorings. The core of the paper then outlines the main challenges that arise from our aim to produce practical tools for a decidedly non-toy language, summarizes our experience in trying to establish the necessary meta-programming infrastructure and gives an implementation overview of our current prototype refactoring tool. Using Haskell as our implementation language, we also offer some preliminary comments on Haskell programming-in-the-large.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

Web Images Video News Maps Gmail more ▾ Sign in

Google

"WebSphere Studio" Java [Search] Advanced Search Preferences

New! View and manage your web history

Web Results **1** - **10** of about **607,000** for **"WebSphere Studio" Java**. (**0.17** seconds)

IBM - Rational Application Developer - Rational Application ...
Helps **Java**™ developers rapidly design, develop, assemble, test, profile and deploy high quality ... Shortens the **Java** learning curve through visual design, ...
www.ibm.com/software/awdtools/studioappdev/ - 23k - Cached - Similar pages

> IBM : developerWorks : Site maintenance
> The IBM developerWorks Web site is currently under maintenance.
> www.ibm.com/developerworks/cn - 9k - Cached - Similar pages
> [More results from www.ibm.com]

IBM releases **WebSphere Studio** 5.0 - **Java** World
WebSphere Studio Application Developer is for building, testing, debugging, and deploying **Java** and J2EE (**Java** 2 Platform, Enterprise Edition) applications. ...
www.**java**world.com/**java**world/jw-09-2002/jw-0927-iw-websphere.html - 42k - Cached - Similar pages

> **WebSphere Studio** Application Developer 4.0 - **Java** World
> IBM's **WebSphere Studio** Application Developer 4.0 IDE can help small and midsize teams speed J2EE (**Java** 2 Platform, Enterprise Edition) application ...
> www.**java**world.com/**java**world/jw-03-2002/jw-0322-iw-websphere.html - 43k - Cached - Similar pages

IBM Redbooks | Using **WebSphere Studio** Device Developer to Build ...
WebSphere Studio Device Developer is the IBM tool for **Java** development for client (embedded) applications. **WebSphere Studio** Device Developer: ...
www.redbooks.ibm.com/abstracts/sg247082.html - 22k - Cached - Similar pages

> IBM Redbooks | Servlet and JSP Programming with IBM **WebSphere ...**
> Running the PDK in VisualAge for **Java** Chapter 15. Developing the PDK using **WebSphere Studio** Appendix A. JSP Tag Syntax Appendix B. Utility Servlet and ...
> www.redbooks.ibm.com/abstracts/sg245755.html - 22k - Cached - Similar pages
> [More results from www.redbooks.ibm.com]

midrange.com -- JAVA400-L -- Calling Program from **Websphere Studio ...**
Subject: Calling Program from **Websphere Studio Java**; From: Jim Mason <JEMason@xxxxxxxxxxxxxx>; Date: Fri, 12 Oct 2001 12:20:44 -0400 ...
archive.midrange.com/**java**400-l/200110/msg00141.html - 17k - Cached - Similar pages

WebSphere Studio Enterprise Developer http://www.ibm.com/software ...
That is why EGL, included with **WebSphere Studio**, is viewed as a path to adopting **Java**; the difficulty of moving a COBOL developer to the object oriented ...
www-306.ibm.com/software/support/rss/websphere/743.xml?rss=s743&ca=rsswebsphere - 35k - Cached - Similar pages

Tool Report: **WebSphere Studio** Profiling Tool
The tool targets applications of all levels of complexity, from simple standalone **java** applications to complex enterprise applications running on multiple ...
www.**java**performancetuning.com/tools/websphereprofiler/ - 20k - Cached - Similar pages

QNX **Java** Environment

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+rename +refactoring SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before October 2004
Terms used: **rename** **refactoring**

Found **68** of **162,233**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 68

Result page: **1** 2 3 4 next

Relevance scale

1 Session 1 (full technical papers): evolution in source code: Challenges of refactoring C programs

Alejandra Garrido, Ralph Johnson
May 2002 **Proceedings of the International Workshop on Principles of Software Evolution IWPSE '02**
Publisher: ACM Press

Full text available: pdf(687.83 KB) Additional Information: full citation, abstract, references, citings, index terms

Refactoring has become a well-known technique for transforming code in a way that preserves behavior. Refactorings may be applied manually, although manual code manipulation is error prone and cumbersome, so maintainers need tools to make automatic refactorings. There is currently extensive literature on refactoring object-oriented programs and some very good tools for refactoring Smalltalk and Java code. Although there is more code written in C or C++ than in any other language, refactoring too ...

Keywords: C programming, preprocessor directives, refactoring

2 Tool support for refactoring functional programs

Huiqing Li, Claus Reinke, Simon Thompson
August 2003 **Proceedings of the 2003 ACM SIGPLAN workshop on Haskell Haskell '03**
Publisher: ACM Press

Full text available: pdf(156.41 KB) Additional Information: full citation, abstract, references, citings, index terms

Refactorings are source-to-source program transformations which change program structure and organisation, but not program functionality. Documented in catalogues and supported by tools, refactoring provides the means to adapt and improve the design of existing code, and has thus enabled the trend towards modern agile software development processes. Refactoring has taken a prominent place in software development and maintenance, but most of this recent success has taken place in the OO and XP co ...

Keywords: Haskell, language-aware programming environments, program transformation, refactoring, semantic editors

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+refactoring +software +code SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before October 2004
Terms used: **refactoring software code**

Found **354** of **162,233**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200   Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown   Relevance scale □ ▭ ▬ ▬ ■

1 Software engineering: Trends in Java code changes: the key to identification of refactorings? ■
Steve Counsell, Youssef Hassoun, Roger Johnson, Keith Mannock, Emilia Mendes
June 2003 **Proceedings of the 2nd international conference on Principles and practice of programming in Java PPPJ '03**
Publisher: Computer Science Press, Inc.
Full text available: pdf(88.06 KB)   Additional Information: full citation, abstract, references, citings, index terms

Changes made to object-oriented (OO) systems over time provide an insight into both design robustness and changes in requirements. When expressed at a *high level* of abstraction, observing trends in changes to code can indicate opportunities for refactoring at the architectural level. In this paper, we empirically investigate the changes made to a set of fifty-two Java library classes over a three year period. The research attempts to support the hypothesis that certain types of changes ma ...

Keywords: Java, changes, code, refactoring

2 Removing false code dependencies to speedup software build processes ■
Yijun Yu, Homy Dayani-Fard, John Mylopoulos
October 2003 **Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research CASCON '03**
Publisher: IBM Press
Full text available: pdf(158.71 KB)   Additional Information: full citation, abstract, references, citings, index terms

The development of large software systems involves a continual lengthy build process that may include preprocessing, compilation and linking of tens of thousands of source code files. In many cases, much of this build time is wasted because of false dependencies between implementation files and their respective header files. We present a graph algorithm and a programming tool that discovers and removes false dependencies among files. We show experimentally that the resulting preprocessed code is ...

3 Session 1 (full technical papers): evolution in source code: Challenges of refactoring C programs ■
Alejandra Garrido, Ralph Johnson
May 2002

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+refactoring +dependent +modules SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before October 2004
Terms used: **refactoring dependent modules**

Found **45** of **162,233**

Sort results by	relevance
Display results	expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 45 Result page: **1** 2 3 next

Relevance scale

1 An Aristotelian understanding of object-oriented programming
Derek Rayside, Gerard T. Campbell
October 2000 **ACM SIGPLAN Notices , Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '00**, Volume 35 Issue 10
Publisher: ACM Press
Full text available: pdf(357.08 KB) Additional Information: full citation, abstract, references, citings, index terms

The folklore of the object-oriented programming community at times maintains that object-oriented programming has drawn inspiration from philosophy, specifically that of Aristotle. We investigate this relation, first of all, in the hope of attaining a better understanding of object-oriented programming and, secondly, to explain aspects of Aristotelian logic to the computer science research community (since it differs from first order predicate calculus in a number of important ways). In both res ...

2 Technical papers: software maintenance: Evolving legacy system features into fine-grained components
Alok Mehta, George T. Heineman
May 2002 **Proceedings of the 24th International Conference on Software Engineering ICSE '02**
Publisher: ACM Press
Full text available: pdf(1.42 MB) Additional Information: full citation, abstract, references, index terms

There is a constant need for practical, efficient, and cost-effective software evolution techniques. We propose a novel evolution methodology that integrates the concepts of features, regression tests, and component-based software engineering (CBSE). Regression test cases are untapped resources, full of information about system features. By exercising each feature with its associated test cases using code profilers and similar tools, code can be located and refactored to create components. These ...

3 Removing false code dependencies to speedup software build processes
Yijun Yu, Homy Dayani-Fard, John Mylopoulos
October 2003 **Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research CASCON '03**
Publisher: IBM Press
Full text available: pdf(158.71 KB) Additional Information: full citation, abstract, references, citings, index

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+abstract:automatically +abstract:refactoring +abstract:chan(SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before October 2004
Terms used: **automatically refactoring changes**

Found **2** of **162,233**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 2 of 2

Relevance scale □ ▨ ▨ ▨ ■

1 Automated method-extraction refactoring by using block-based slicing ■

Katsuhisa Maruyama
May 2001 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2001 symposium on Software reusability: putting software reuse in context SSR '01**, Volume 26 Issue 3
Publisher: ACM Press
Full text available: pdf(174.09 KB) Additional Information: full citation, abstract, references, index terms

Refactoring improves the design of existing code but is not complex to do by hand. This paper proposes a mechanism that automatically refactors methods of object-oriented programs by using program slicing. To restructure a method without changing its observable behavior, the mechanism uses block-based slicing that does not extract the fragments of code from the whole program but from the region consisting of some consecutive basic-blocks of the program. A refactoring tool implementing the m ...

2 Evolving Object-Oriented Designs with Refactorings ■

Lance Tokuda, Don Batory
October 1999 **Proceedings of the 14th IEEE international conference on Automated software engineering ASE '99**
Publisher: IEEE Computer Society
Full text available: Publisher Site Additional Information: full citation, abstract, citings

Refactorings are behavior-preserving program transformations that automate design level changes in object-oriented applications. Our previous research established that many schema transformations, design patterns, and hot-spot meta-patterns are automatable. This research evaluates whether refactoring technology can be transferred to the mainstream by restructuring non-trivial C++ applications. The applications that we examine were evolved manually by software engineers. We show that an equivalen ...

Keywords: design evolution, refactorings, patterns

Results 1 - 2 of 2

Terms of Usage Privacy Policy Code of Ethics Contact Us